

## Egen Coding Challenge – Level 1

Restful web service for User Management application using

- Spark Java
- MongoDB

### Services developed

#### **1. createUser**

- Takes input as JSON.
- Creates the user if he is not already available in the data store.

#### **2. getAllUsers**

- Gives the list of all users that are in the data store

#### **3. updateUser**

- Takes input as JSON and Finds the user from the data store
- if found, updates the required fields
- if not found, sends back a 404 saying user not found

#### **4. getUserByEmail**

- Finds and gives the user details based on email.

### Tests

- Developed controller integration tests using Junit

### Assumptions

- **Email** attribute of user is considered as unique.
- User id is auto-generated by ObjectId

### Steps to run the project

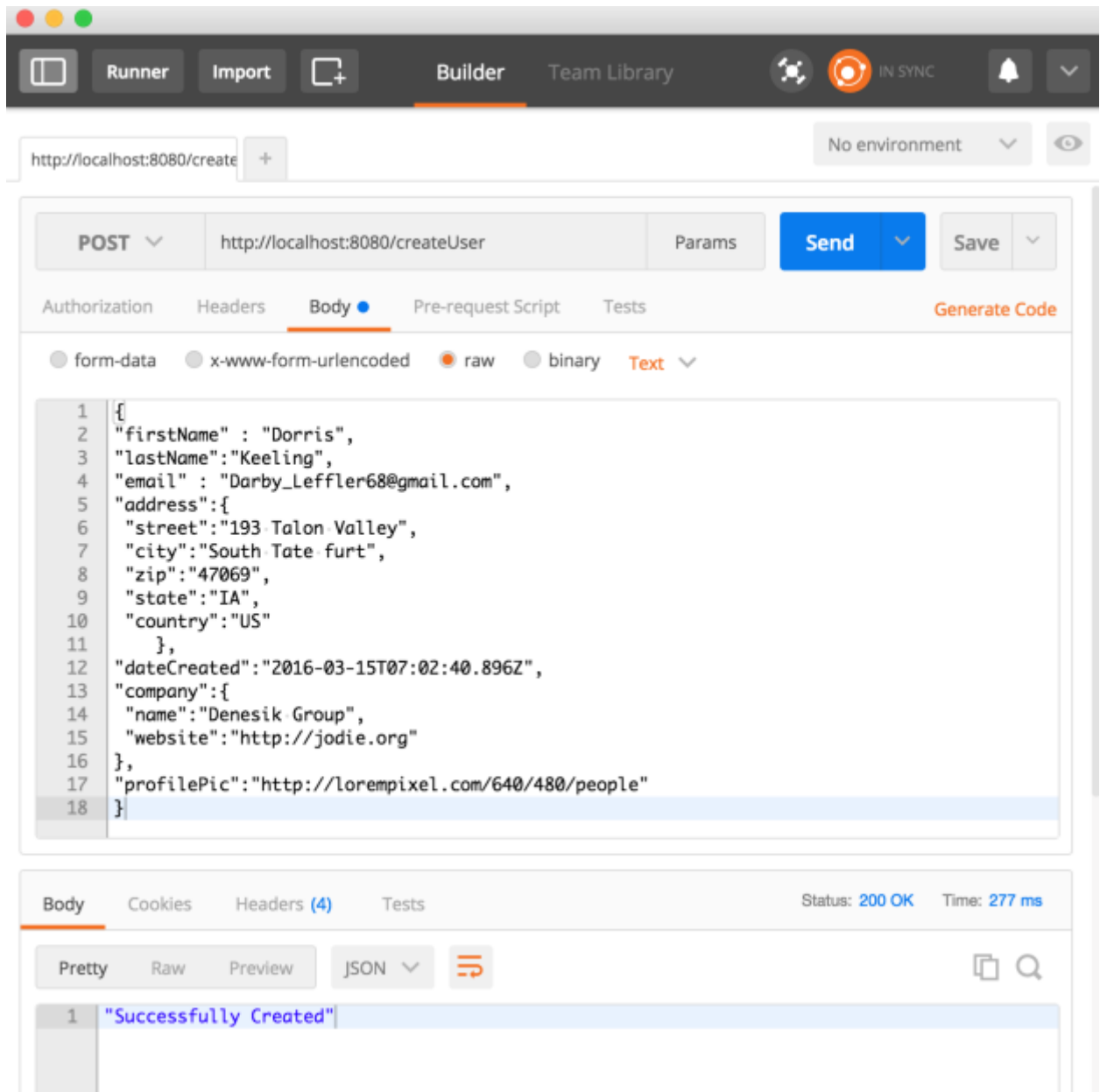
- Install Maven and MongoDB
- Get all the dependencies in Maven
- To execute, run main method under com.egen.restfulapp.Main

## Tools and Technologies used

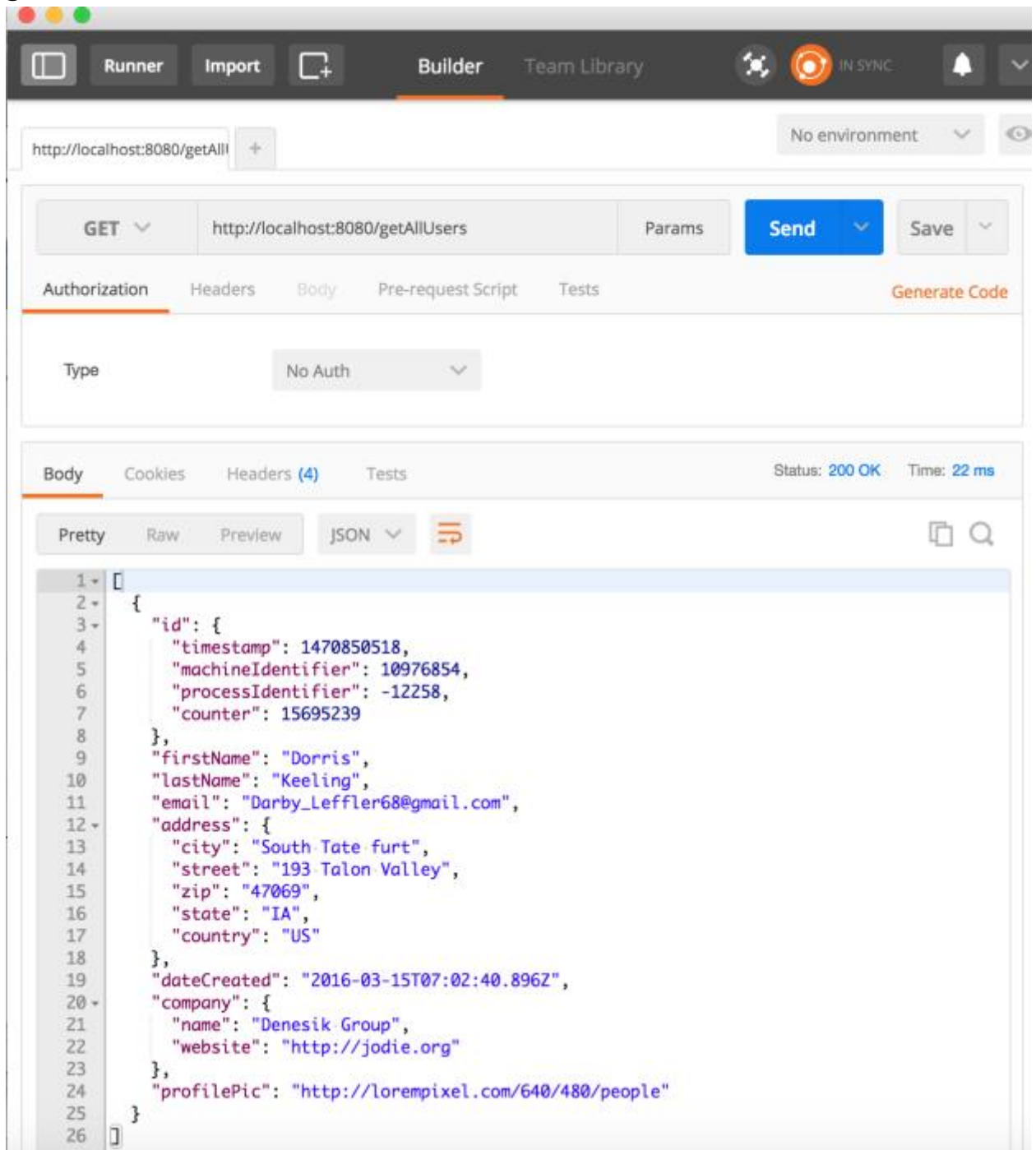
- Spark Java, MongoDB, Morphia, SLF4J and JUnit
- Eclipse, Postman and Robomongo

## Project Execution Screenshot's

### 1. createUser



## 2. getAllUsers



The screenshot displays a REST client interface with the following components:

- Runner** and **Builder** tabs are active.
- The URL bar shows `http://localhost:8080/getAll`.
- The **GET** method is selected for the endpoint `http://localhost:8080/getAllUsers`.
- The **Authorization** tab is selected, showing **Type** as **No Auth**.
- The **Body** tab is selected, showing the response in **JSON** format.
- The status is **200 OK** and the time taken is **22 ms**.

The response body is a JSON array containing one user object:

```
1 [
2   {
3     "id": {
4       "timestamp": 1470850518,
5       "machineIdentifier": 10976854,
6       "processIdentifier": -12258,
7       "counter": 15695239
8     },
9     "firstName": "Dorris",
10    "lastName": "Keeling",
11    "email": "Darby_Leffler68@gmail.com",
12    "address": {
13      "city": "South Tatefurt",
14      "street": "193 Talon Valley",
15      "zip": "47069",
16      "state": "IA",
17      "country": "US"
18    },
19    "dateCreated": "2016-03-15T07:02:40.896Z",
20    "company": {
21      "name": "Denesik Group",
22      "website": "http://jodie.org"
23    },
24    "profilePic": "http://lorempixel.com/640/480/people"
25  }
26 ]
```

### 3. UpdateUser

The screenshot displays a REST client interface with the following components:

- Top Bar:** Includes tabs for Runner, Import, Builder (active), and Team Library. It also shows a status indicator 'IN SYNC' and a notification bell.
- Address Bar:** Shows the URL 'http://localhost:8080/updat' with a dropdown menu.
- Request Configuration:** The method is set to 'PUT' and the URL is 'http://localhost:8080/updateUser'. There are buttons for 'Send' and 'Save'.
- Request Body:** The 'Body' tab is selected. The content type is set to 'raw'. The JSON body is as follows:

```
1 {  
2   "firstName": "AnkithJainUpdate",  
3   "lastName": "Keeling",  
4   "email": "Darby_Leffler68@gmail.com",  
5   "address": {  
6     "city": "South Tate furt",  
7     "street": "193 Talon Valley",  
8     "zip": "47069",  
9     "state": "IA",  
10    "country": "US"  
11  },  
12  "dateCreated": "2016-03-15T07:02:40.896Z",  
13  "company": {  
14    "name": "Denesik Group",  
15    "website": "http://jodie.org"  
16  },  
17  "profilePic": "http://lorempixel.com/640/480/people"  
18 }
```
- Response:** The 'Body' tab is selected. The status is '200 OK' and the time is '20 ms'. The response body is:

```
1 "200: Succesfully updated user"
```

#### 4. getUserByEmail

The screenshot shows a REST client interface with the following components:

- Top Bar:** Includes tabs for Runner, Import, Builder (active), and Team Library. It also shows a status 'IN SYNC' and a bell icon.
- URL Bar:** Displays the URL `http://localhost:8080/getUs` with a dropdown menu.
- Request Section:**
  - Method:** GET
  - URL:** `http://localhost:8080/getUserByEmail/Darby_Leffler68@gmail.com`
  - Params:** A button to add parameters.
  - Buttons:** Send (blue), Save (dropdown).
  - Tabs:** Authorization, Headers, Body (selected), Pre-request Script, Tests.
  - Generate Code:** A button to generate code from the request.
  - Type:** No Auth (dropdown).
- Response Section:**
  - Tabs:** Body (selected), Cookies, Headers (4), Tests.
  - Status:** 200 OK
  - Time:** 8 ms
  - Format:** JSON (dropdown), Pretty (selected), Raw, Preview.
  - JSON Content:**

```
1 {
2   "id": {
3     "timestamp": 1470850518,
4     "machineIdentifier": 10976854,
5     "processIdentifier": -12258,
6     "counter": 15695239
7   },
8   "firstName": "AnkithJainUpdate",
9   "lastName": "Keeling",
10  "email": "Darby_Leffler68@gmail.com",
11  "address": {
12    "city": "South-Tate-furt",
13    "street": "193-Talon-Valley",
14    "zip": "47069",
15    "state": "IA",
16    "country": "US"
17  },
18  "dateCreated": "2016-03-15T07:02:40.896Z",
19  "company": {
20    "name": "Denesik Group",
21    "website": "http://jodie.org"
22  },
23  "profilePic": "http://lorempixel.com/640/480/people"
24 }
```

## Tests

